Research Report

Psychometric and Cognitive Analysis as a Basis for the Design and Revision of Quantitative Item Models

Edith Aurora Graf

Stephen Peterson

Manfred Steffen

René Lawless

Research & Development

December 2005
RR-05-25

Psychometric and Cognitive Analysis as a Basis for the Design and Revision of Quantitative Item Models

Edith Aurora Graf and Stephen Peterson
ETS, Princeton, NJ

Manfred Steffen
CTB/McGraw-Hill, Monterey, CA

René Lawless
ETS, Princeton, NJ

December 2005

As part of its educational and social mission and in fulfilling the organization's nonprofit charter and bylaws, ETS has and continues to learn from and also to lead research that furthers educational and measurement research to advance quality and equity in education and assessment for all users of the organization's products and services.

ETS Research Reports provide preliminary and limited dissemination of ETS research prior to publication. To obtain a PDF or a print copy of a report, please visit:

http://www.ets.org/research/contact.html

Abstract

We describe the item modeling development and evaluation process as applied to a quantitative assessment with high-stakes outcomes. In addition to expediting the item-creation process, a model-based approach may reduce pretesting costs, if the difficulty and discrimination of model-generated items may be predicted to a predefined level of accuracy. The development and evaluation of item models represents a collaborative effort among content specialists, statisticians, and cognitive scientists. A cycle for developing and revising item models that generate items with more predictable statistics is described. We review the goals of item modeling from different perspectives and recommend a method for structuring families of models that span content and generate items with more predictable psychometric parameters.

Key words: Item model, item family, automatic item generation, cognitive analysis, mathematics assessment

Acknowledgements

This report combines two documents, the original versions of which were included in a set of symposium papers presented at the International Association for Education Assessment conference in June 2004. The session was chaired by René Lawless. We would like to thank Jutta Levin, Mary Morley, Sheng Wang, and Sarah Ohls for their help with designing the item models, generating instances, and coordinating this research; Isaac Bejar for his insight and support; Sandip Sinharay for his advice; Rochelle Robert for her feedback on an earlier version of this paper; Dele Kuku for her efforts in reformatting this document from its original version; Vicky Pszonka for her assistance with formatting and locating references, and Kim Fryer for coordinating the copy-editing. Finally, we would like to thank Liz Marquez, Randy Bennett, Mary Enright, Dan Eignor, and Sarah Ohls for their comments. We appreciate their contributions, and they are not responsible for any errors contained herein.

Table of Contents

Introduction

A model-based approach to generative testing has the potential for enhancing both efficiency and validity (Bejar et al., 2002). We begin this paper by defining background terminology and giving simple examples of item models; this is followed by a review of the advantages of using a model-based approach to test development. Design practices that address security issues are also discussed.

Next, we describe an effort to develop a set of model-based, automatically generated mathematics items with predictable statistical characteristics. When using a model-based approach, the greatest gains in efficiency are conferred when it is possible to predict the statistical characteristics of the items with some degree of accuracy, since this may eliminate the need to collect statistics for each item individually (Steffen, Graf, Levin, & Robin, 2005). There is a second reason why the accurate prediction of item statistics is a useful goal: It requires a thorough understanding of the features that contribute to item complexity, which is valuable information for test developers and researchers in the long run.

Generating items with predictable statistics can be challenging and less than straightforward. Item modeling depends upon a thorough analysis of the underlying assessment domain (Bejar, 2002). Conducting such an analysis is necessary to generate a set of items that addresses a specified set of skills, but in complex domains, it is only the first step in developing a set of items at predictable levels of difficulty and discrimination. In this paper, we define a process for developing and revising sets of automatically generated mathematics items with more predictable statistical characteristics. The item-model development process is iterative and represents a collaborative effort on the part of test developers, statisticians, and cognitive scientists. The process was developed for use in the context of an assessment with high-stakes outcomes; however, some steps may be applicable to low-stakes assessment contexts as well.

Item Models: Background and Terminology

Hively, Patterson, and Page (1968) used the term *item forms* to refer to classes of items that assess the same subject matter and share a set of explicitly defined formats and mathematical relationships. They developed a number of item forms to characterize classes of basic arithmetic items. The term *item model* was introduced by LaDuca, Staples, Templeton, and Holzman (1986) to define classes of *content-equivalent* items. An item defined by an item model is sometimes referred to as an *instance* (Bejar et al., 2002).

Item models may be broadly defined or more narrowly defined. For example, “Find the sum of two positive common proper fractions, each of which has a denominator less than or equal to 10,” is a verbal description of a broadly defined quantitative item model. The instances 2/4 + 2/3 = ?, 3/8 + 7/10 = ?, 1/4 + 2/5 = ?, 1/7 + 3/7 = ?, and 3/6 + 4/6 = ? are all included in this model. “Find the sum of two reduced positive common proper fractions with different denominators, each of which has a denominator less than or equal to 10,” is a more narrowly defined model that describes a smaller item space. This item model describes a set of instances that constitutes a proper subset of the set described by the broader model. The instances 3/8 + 7/10 = ? and 1/4 + 2/5 = ? are included in this model, but the instances 2/4 + 2/3 = ?, 1/7 + 3/7 = ?, and 3/6 + 4/6 = ? are not. Both example item models are represented in Table 1.

Table 1

Two Example Item Models

Description of model	Model template	Variables and constraints	Examples
Find the sum of two positive common proper fractions, each of which has a denominator less than or equal to 10.	*Num1*/*Den1* + *Num2*/*Den2* = ?	*Num1* is an integer s.t. $1 \leq Num1 \leq 9$ *Den1* is an integer s.t. $2 \leq Den1 \leq 10$ *Num2* is an integer s.t. $1 \leq Num2 \leq 9$ *Den2* is an integer s.t. $2 \leq Den2 \leq 10$ *Num1*/*Den1* < 1 *Num2*/*Den2* < 1 *Key = Num1*/*Den1* + *Num2*/*Den2*	2/4 + 2/3 = ? 3/8 + 7/10 = ? 1/4 + 2/5 = ? 1/7 + 3/7 = ? 3/6 + 4/6 = ?
Find the sum of two reduced positive common proper fractions with different denominators, each of which has a denominator less than or equal to 10.	*Num1*/*Den1* + *Num2*/*Den2* = ?	*Num1* is an integer s.t. $1 \leq Num1 \leq 9$ *Den1* is an integer s.t. $2 \leq Den1 \leq 10$ *Num2* is an integer s.t. $1 \leq Num2 \leq 9$ *Den2* is an integer s.t. $2 \leq Den2 \leq 10$ *Num1*/*Den1* < 1 *Num2*/*Den2* < 1 *Key = Num1*/*Den1* + *Num2*/*Den2* *Num1* and *Den1* are relatively prime *Num2* and *Den2* are relatively prime *Den1* is not equal to *Den2*	3/8 + 7/10 = ? 1/4 + 2/5 = ?

Note. Variables are shown in italics.

Although quantitative item models may be captured on paper, they can also be programmed into software. The same software may be used to automatically generate instances specified by the model (Singley & Bennett, 2002). Components of an item model that vary across instances are represented as *variables*. Statements that restrict the values that variables are

permitted to assume are called *constraints.* The variables and constraints used in the fraction addition examples are shown in the third column of Table 1. Both models use the same set of variables: four integer variables that represent the numerators and denominators of each of the two addends (*Num1*, *Den1*, *Num2*, *Den2*), and a fraction variable that represents the key (*Key*). Similarly, both models share three constraints: ($Num1/Den1 < 1$, $Num2/Den2 < 1$, and $Key = Num1/Den1 + Num2/Den2$). The first two constraints ensure that the addends are less than 1. The last constraint defines the key. The narrower model includes three additional constraints: *Num1* and *Den1* are relatively prime, *Num2* and *Den2* are relatively prime, and *Den1* is not equal to *Den2*. The first two additional constraints ensure that the addends are reduced fractions. The third additional constraint specifies that the denominators of the two addends must not be equal.

As mentioned earlier, the narrower model describes a set of instances that constitutes a proper subset of the set described by the broader model. The two models are very similar; in fact, the only difference between them is that the narrower model includes three additional constraints. Similar item models are often grouped into item model *families*. Often, item models in the same family share variables, constraints, or both. Item models in the same family may share a common mathematical structure but vary with respect to their surface features, or vice-versa.

Advantages of Using an Item Modeling Approach in a High-Stakes Assessment Program

Using a model-based approach in testing programs with high-stakes outcomes may enhance efficiency and validity; security issues may be addressed through appropriate design practices.

Efficiency. Because an item model can be used to automatically generate a large number of instances, a model-based approach to item development is a potentially economical solution to meeting item demands (Bejar et al., 2002). Item modeling may enhance efficiency in at least two different ways. First, automatic item generation has the potential to lower the cost of item development. While it is more time-consuming to develop an item model than an item, the development cost per unit item may be lower, assuming the model generates a large number of instances.

Second, a model-based approach may lower pretesting costs. Ideally, quantitative item models are based on an underlying problem structure or schema (Singley & Bennett, 2002). If it is possible to predict the item parameter estimates of an instance based on its item model

structure, it may not be necessary to calibrate every instance individually; rather, information about how difficulty is related to item model variables may be used to predict item parameter estimates for instances in advance (e.g., Bejar, 1993; Bejar, 1996; Bejar et al., 2002; Bejar & Yocom, 1991; Embretson, 1999). Enright, Morley, and Sheehan (2002) used item models to assess the impact of item design features on difficulty and discrimination. For rate word problems, feature variation accounted for 90% of the variance in difficulty and 50% of the variance in discrimination; for probability problems, feature variation accounted for 61% of the variance in difficulty but did not explain the variance in discrimination. This suggests that it is possible to use item features to successfully predict item difficulty. As noted by Enright et al., however, the feasibility of using item features to predict difficulty should be explored for many other mathematics topics as well. More recently, there have been additional evaluations of the feasibility of predicting item parameters of instances generated from quantitative models (Sinharay & Johnson, 2005; Steffen et al., 2005).

Generative response modeling and its relationship to validity. Generative response modeling (Bejar, 1993; Bejar, 1996; Bejar & Yocom, 1991) is an approach that relates generative principles to psychometric properties. A generative response model not only specifies how an instance should be produced, it also specifies how the generating features are related to item difficulty and discrimination. Ideally, an item model is also a generative response model—it should specify the relationships between its features and the psychometric properties of the instances it generates. In the context of our discussion about quantitative item models, this means describing how the variables and constraints specified in the model relate to the difficulty and discrimination of the instances.

Irvine (2002) makes a distinction between *radical* and *incidental* item elements. In an item model, a radical is a variable that affects the psychometric characteristics of the instances, and an incidental is a variable that has no detectable effect on the psychometric characteristics. A numeric variable that influences the computational complexity of an item is most likely radical, but a variable that is replaced by a random text string (for example, the name of a person in a word problem) is most likely incidental. A constraint may be used to mediate variable effects. For example, perhaps it is expected that computations involving the numeric variable *N* will be comparable in difficulty, unless *N* assumes a value divisible by 10. If a constraint is written so

that *N* cannot be divisible by 10, it is expected that the variable *N*, in combination with the constraint, will be incidental rather than radical.

Returning to the fraction example, a test developer may anticipate that adding fractions with different denominators will be more difficult for an examinee than adding fractions with like denominators; in other words, the relationship between the denominators is a radical element. In the context of the broader model, this implies a prediction: On average, instances with different denominators will likely be more difficult than instances with like denominators. In the context of the narrower model, this intuition is operationalized as a constraint. The narrower model is designed to generate instances that vary less in difficulty; at least, any variation in difficulty is not due to whether or not the addends share the same denominator since they never will by definition. Although the broad and narrow models are defined differently, both may be considered generative response models, because we have incorporated expectations for how generating principles may influence difficulty.

Bejar (1993) and Bejar and Yocom (1991) have argued that a careful accounting of the item model features that contribute to item difficulty lends validity to an assessment, and that generative response modeling provides a systematic framework in which to explore the determinants of item difficulty. In the fraction examples, we assume that adding fractions with different denominators is more difficult than adding fractions with the same denominator. If this is true, the level of performance on instances in the former category should on average be lower than the level of performance on instances in the latter category. To the extent that this feature is a relatively important determinant of difficulty, the variability in performance among instances in the broadly defined model should significantly exceed the variability among the instances in the narrowly defined model. If this hypothesis is not confirmed, it allows for the possibility that there are other variables that were not considered that are relatively more important for explaining difficulty. There are many other variables that might be considered in defining fraction addition item models. Hively et al. (1968) distinguished among numerous fraction addition item forms with different properties, including addends with the same denominators, relatively prime denominators, one denominator a multiple of the other, and denominators with a common factor. Further, they characterized forms with mixed numbers, forms with sums less than 1, and so on. Any one of these forms could be represented as an item model by including

the appropriate subset of constraints. For example, the constraint *Key* < 1 could be added to a model so that it would only generate instances with sums less than 1.

The preceding discussion suggests two alternative (and possibly complementary) methods for using item models to identify factors that affect difficulty. The first method is to design broadly defined item models, but also to develop corresponding cognitive and measurement models that are sufficiently complex to explain the psychometric variability among the instances. The second method is to constrain the item models so that the performance characteristics of the instances within each model are relatively uniform. Bejar (2002) and Bejar and Yocom (1991) characterize the second method as a special case of the first method. Instances with predictable variation in their psychometric parameters are called *variants*; instances with similar psychometric parameters are called *isomorphs* (Bejar, 2002). Another characterization of the second method is that radical variables are used across models, but only incidental variables are used within models. Meisner, Luecht, and Reckase (1993) investigated the statistical comparability of mathematics instances that were generated from the same *algorithm* (an algorithm is an item model that generates instances that share a common mathematical structure). They found that many algorithms generated instances with similar statistics, but that a few did not.

Whether an item model is designed to generate variants or isomorphs, an explanatory mechanism that accounts for the psychometric characteristics of the instances supports validity evaluation, because it makes it possible to demonstrate empirically an understanding of the features that contribute to item difficulty and discrimination. Ideally, any radical elements included in an item model are relevant to the target skill. If instances of an item model appropriately measure the target skill, it should be possible to keep the level of unexplained performance variability to a reasonable minimum. In the case where an item model is designed to generate isomorphs, it should be possible to keep the psychometric parameters constant, if the item model variables that influence them are thoroughly understood (Embretson & Gorin, 2001).

In order to assist the item model design process, a framework such as evidence-centered design (Mislevy, Steinberg, & Almond, 2002) may be used. This approach to assessment design is founded on the assumption that tasks, scoring, and the reasoning that links them should be based on a thoughtful specification of the construct that the assessment is intended to measure. The interrelationships among the requisite student proficiencies for an assessment are

represented in a proficiency model. Shute, Graf, and Hansen (2005) used the evidence-centered design approach to develop a proficiency model for a middle school–level mathematics unit on arithmetic, geometric, and other common progressions. They then authored item models to correspond to the skills represented in the proficiency model. Although an approach like evidence-centered design can guide the item model design process, it does not provide a guarantee that all performance variability among the instances will be accounted for: The expectations for the psychometric parameters of the instances must be empirically evaluated.

Test security. If it is possible to design an item model that generates instances with constant item parameters, the instances are effectively exchangeable. If an instance can be replaced with another instance of the same model, this will reduce the overexposure of administered items and enhance test security. There is reason for caution, however. Since isomorphic instances are based on a common problem structure, it is possible that they are more recognizable than discretely authored items. Instances that are programmed on a computer may share incidental but systematic relationships among the modeled components, some of which may be construct-irrelevant. If such relationships are discernable, construct-irrelevant, and correlated with the key, this presents a security concern as increasing numbers of instances from a model are administered. This concern is expressed as the degree to which item models may be *coachable*.

Fortunately, some of this concern may be alleviated through the careful design of item models and may even be counteracted. Morley, Bridgeman, and Lawless (2004) designed models that generated isomorphs and models that generated *appearance variants*. The former models generated instances that were similar in difficulty and shared a common mathematical structure; the latter models generated instances that looked similar with respect to their surface features, but varied in terms of their mathematical structure (i.e., it would be necessary to use alternate mathematics procedures to derive a correct response). In the study by Morley et al., all participants were administered the same posttest (comprised of the same set of base items), but the composition of the pretests they were administered was different. Each base item on the posttest was paired with a corresponding item on a pretest. There were three types of pairs: (a) isomorphic pairs, (b) appearance variant pairs[1], and (c) difficulty-matched pairs (these were items that were similar in difficulty and measured the same general mathematical skills, but required different solution strategies and were completely different superficially). Across the

pretest and the posttest, some participants saw isomorphic pairs and matched pairs only; the other participants saw all three types: isomorphic pairs, matched pairs, and appearance variant pairs. Morley et al. found that participants performed better on items for which they had seen isomorphs, but the inclusion of appearance variants on the pretest diminished this effect. They concluded that including both types of modeled instances on tests might discourage examinees from attending to the construct-irrelevant features of item models. A more in-depth study of whether long-term coaching may have an impact on item model security has yet to be conducted.

Now that we have defined the necessary background terminology and discussed some of the advantages of an item modeling approach to assessment design, we describe an effort to develop a set of models, and a process by which item models may be developed and revised.

Project Overview

In this paper, we describe a process that may be used to design and revise quantitative item models. The process was developed in the context of a research project, the goal of which was to design a set of item models that would generate psychometrically equivalent (isomorphic) multiple-choice instances for use in unscored quantitative sections of the Graduate Record Examination® (GRE®). Item models can be designed for either high-stakes or low-stakes applications, but here we focused on item models designed for use in assessments with high-stakes outcomes. The development effort that was the basis for this report is described in Steffen et al. (2005). Development spanned four mathematical content areas (linear inequalities, probabilities, remainders, and quadrilateral perimeters). In each of these content areas, a model was developed at each of four levels of difficulty (easiest, moderately easy, moderately hard, and most difficult), making sixteen models. Ten multiple-choice instances were generated from each of the 16 models and inserted into unscored test sections. Although the sections were administered internationally, the analysis sample consisted of U.S. citizens for whom English is the best language. Examinee responses were scored as correct or incorrect, and the response scores were fitted to a three-parameter logistic (3PL) IRT model (Birnbaum, 1968). Some of the models generated instances with similar item parameter estimates, but others generated instances with highly variable item parameter estimates.

We will first describe the item model development process from the perspective of a test developer who participated in the process. Next, we will describe an approach to identify some of the cognitive and mathematical features that may account for the observed psychometric

variability among the instances of the item models, and how to revise the models to reduce the variability. Finally, we will recommend an approach for developing isomorphic instances while maintaining sufficient variability in item content.

Developing Item Models: A Test Developer's Perspective

This section describes item modeling from the perspective of a test developer responsible for creating multiple-choice questions for tests with high-stakes outcomes. The following issues are covered

- How can test developers use item modeling to create large numbers of test questions that assess the same mathematical concepts?
- How can test developers use item modeling to create large numbers of test items at predictable levels of difficulty and discrimination?
- What are the efficient processes that can be used for item model creation, item model review, and item (instance) review?
- What are the challenges associated with item modeling?

When the decision has been made to use item models in a test with high-stakes outcomes, a process for the development of models must be established.

The Item Model Development Process: An Overview

After the decision to model items has been made, test developers choose items and types of items that are most suitable for item modeling. Groups of related items are examined and analyzed, as part of a preliminary construct analysis. It is not a requirement, but often one or more exemplar items are chosen as the basis for an item model. Such an item is called a *source*, or *base* item. A candidate source item should be evaluated in terms of its mathematical and cognitive properties, to ascertain whether it lends itself to rendering many variations. If the number of possible variations is too small, then that item may not be a suitable source item because it is not cost effective to design a model that generates only a small number of instances. A source item is often selected in accordance with particular psychometric criteria. This is possible because source items are usually selected from a pool of previously pretested items for which comprehensive item-level statistics are available.

An item model should be designed to capture the underlying mathematical structure implicit in the source item. The incidental features of the source item may be varied across the

instances of the model, but mathematical relationships in the source item that are considered important to retain are constrained in the item model. A test developer begins by deciding which parts of a model should be varied across instances and what the relationships among variables should be. Usually the model is sketched out on paper first; later the variables and constraints may be programmed into an automatic item generation program.

An important aspect of multiple-choice item model development is to capture students' approaches to solving problems and to represent common misconceptions among the options. Since the distribution of examinee responses across options is usually available for source items, popular misconceptions can often be identified. For example, suppose the stem of a source item for a multiple-choice fraction addition model is 1/3 + 2/7 = ?, and a large proportion of examinees choose the option 3/10. The test developer may decide to incorporate this misconception into the corresponding item model by representing one of the distractors as the sum of the numerators divided by the sum of the denominators. This ensures that an option that represents this misconception will appear in every instance generated by the model.

Once the model has been developed, one or more test developers review it to ensure that it is both correct and consistent with the goals of the construct analysis, reviewing the model itself as well as samples of the instances it generates. At this point, the developer edits the model in accordance with reviewers' suggestions; additional cycles of reviewing and editing may be warranted. When a model is complete, a sample of its instances is automatically generated for use in an unscored section. Content specialists review each instance for accuracy and clarity. Minor formatting changes are often made directly to the instances; more substantial changes may require that the model is changed and that a new sample of instances is generated.

Once the model has been approved and the instances are finalized, the instances are assembled into unscored sections and administered. After the raw response data is collected, statistics are computed, including classical item statistics and item response theory (IRT) parameter estimates for each of the modeled instances. As part of a retrospective cognitive analysis, we use these statistics to hypothesize the influences of various item features. In this analysis, we attempt to relate the differences among instance statistics to differences among instance features (i.e., stem or distractor differences). The results of this analysis are used either to revise the item model or simply to explain any significant statistical variability among the generated instances. By treating each item model revision as an experiment and by exploring

how modeled distractors influence difficulty, we develop an enhanced understanding of students' interpretations, strategies, and performance. The item model development and analysis process is summarized in Figure 1.

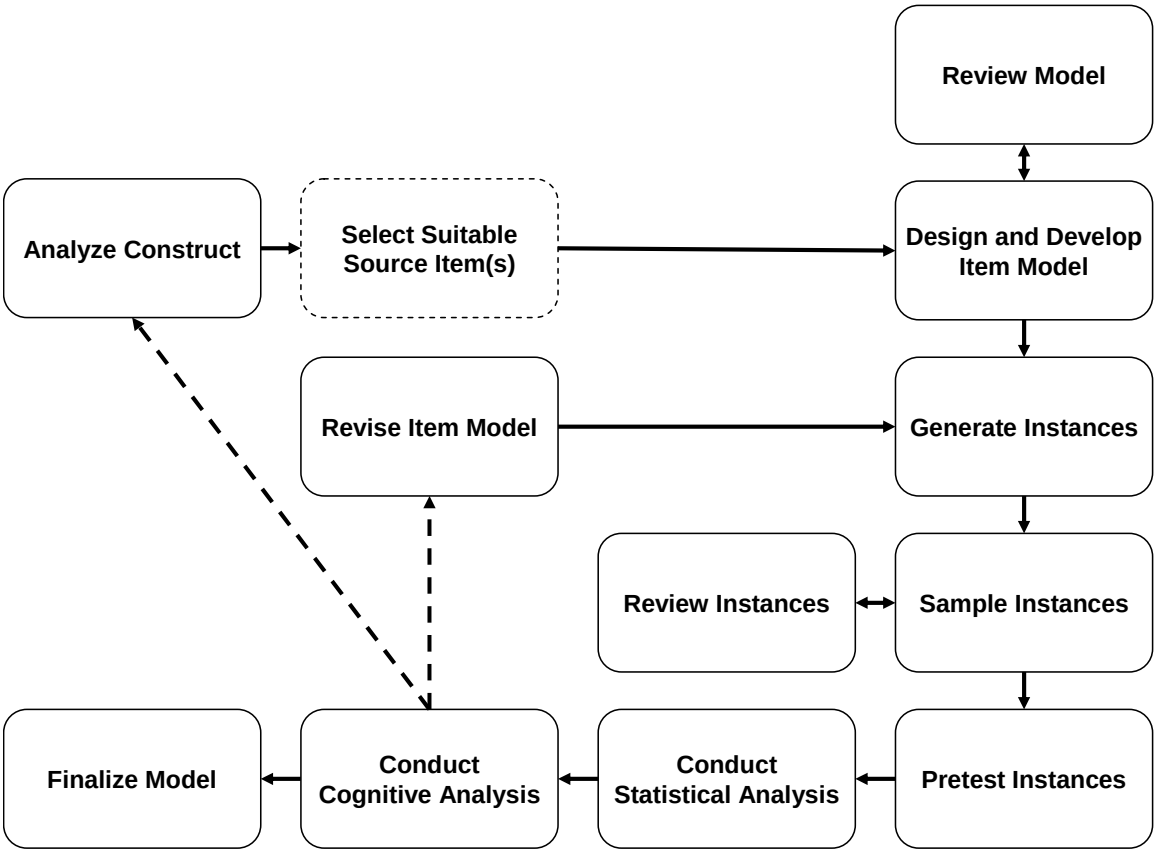

Figure 1. **Item model development and analysis loop.**

Although the process described here is workflow oriented, it shares many parallels with Embretson's cognitive design system approach (e.g., see Embretson, 1999; Embretson & Gorin, 2001). The initial construct analysis is analogous to the develop-cognitive-model phase of Embretson's cognitive-design system approach. In this context, developing the cognitive model means making explicit the cognitive variables that are central to the source item(s), so that an item model that generates isomorphic instances may be constructed. During an initial construct analysis, a team of test developers discusses the appropriate content, reviews similar items in the operational pool, analyzes the source item(s), and develops a model designed to generate isomorphic instances.

The Item Model Development Process: An Example

Selecting or inducing a suitable source item. The first step in the item modeling process is to review large numbers of items and find items that may generate large numbers of similar instances. Often, a single source item serves as the basis for the design of an item model. Sometimes, the item is modified before it is modeled. Occasionally, features from several different items are combined, and a source item is *induced.* The example here uses a source item involving an income tax scenario (see Figure 2).

> In a certain state, for taxable incomes over $20,000, income taxes are calculated as 9 percent of the first $20,000 of taxable income plus 15 percent of the amount greater than $20,000. If the taxes calculated for a certain taxable income were $2.100. what was the taxable income?

Figure 2. **Source item.**

Analysis has shown that about 60% of GRE General Test examinees correctly solve problems of this type, and scores on items of this type correlate well with the rest of the quantitative exam. This item is therefore a good source for an item model of medium difficulty. We will next create an item model designed to generate instances that assess the same mathematical content and have similar psychometric properties.

What is the mathematical content being assessed? This item assesses the mathematical concepts of graduated rates, calculations with percentages, and algebraic manipulation. In this example, the scenario concerns taxes; for the model we retain the surface features and variabilize the values in the problem (see Figure 3).

> In a certain state, for taxable incomes over $***y***, income taxes are calculated as ***r*** percent of the first $***y*** of taxable income plus ***t*** percent of the amount greater than $***y***. If the taxes calculated for a certain taxable income were $***w***. what was the taxable income?

Figure 3. **Item model based on the source item in Figure 1.**

Multiple instances from this model can be generated by replacing the variables with appropriate values. There may be several ways of representing this item, but for the purpose of this model we use variables *r*, *y*, *t*, and *w* to set up and solve the equations as shown in Figure 4.

Let α be the taxable income, $x = \frac{r}{100}$, and $z = \frac{t}{100}$,

then $xy + z(\alpha - y) = w$

Solve for α, $\alpha = \frac{w - xy + zy}{z}$

Figure 4. **Equations used for the purpose of finding appropriate numbers for model.**

Modeling the stem. In creating the model for a real-life scenario such as this, it is important to ensure that the values for the variables are realistic, the answers are reasonable, and the grammar in the stem is correct. In Figure 5, we define the variables *r*, *y*, *t*, and *w* and set the constraints.

r is an integer such that $0 < r \leq 10$.

y is an integer divisible by 5,000 such that $10{,}000 \leq y \leq 30{,}000$.

t is an integer such that $11 \leq t \leq 15$.

$x = \frac{r}{100}$, and $z = \frac{t}{100}$

w is an integer divisible by 1,000, such that $xy < w \leq 10{,}000$.

Then the key is $\alpha = \frac{w - xy + zy}{z}$, α should be an integer.

Figure 5. **Establishing variables and constraints.**

Modeling the distractors. One of the challenging parts of building a model to generate multiple-choice instances is developing formulas for the distractors. As we will see, the cognitive analysis shows that choice and placement of the distractors are often key factors that determine the difficulty of the item. The presence or absence of attractive distractors, usually representative

of common misconceptions, can greatly affect the examinees' responses to an item. Also, the distractors have to make sense in the context of the instance.

The instances also have to adhere to the conventions of a multiple-choice exam; if the distractors and the key are numeric they should be listed in ascending or descending order. The key should appear in different positions across the various instances so that the model has a lower risk of being coachable. Example formulas to represent distractors for the tax scenario include:

- $\alpha + 10{,}000$
- $\alpha - 10{,}000$
- $\frac{w + xy + zy}{z}$
- $xy + w$

Can we vary the scenario? The surface features of a source item can be varied to generate more instances. In this case, another problem type involves sales commissions (see Figure 6).

> At a certain company, commissions on total sales over $***y*** are calculated as ***r*** percent of the first $***y*** of sales plus ***t*** percent of the amount of sales greater than $***y***. If a salesperson's commission was $***w***, what were the salesperson's total sales?

Figure 6. **Item with varied surface features.**

Note that the mathematical structures for the models in Figure 3 and Figure 6 are the same. We give the examinees numbers for *r*, *y*, *t*, and *w* and ask them to set up and solve the same equation as above. As before, we adjust the values of the variables and distractors as well as the grammar, to make sense in the context of the sales commission problem.

Item modeling as a collaborative effort. The process for item modeling is collaborative. Teams of test developers work together to choose the source items, to create and review the models, to discuss the choice of distractors, and to review the instances that will be used on the tests. In our experience, a team of three to four people is the ideal size for the efficient creation and review of the models.

Once the instances are generated and reviewed, they are placed in unscored sections. The sections are administered and statistics are collected. Cognitive scientists and test developers then study the statistics of all the instances of a model to see if the instances preserve the difficulty levels of the source items. If the difficulty varies, we try to determine which factors are causing the variability. After the cognitive analysis, the models are often revised or modified and new instances are generated to incorporate the features uncovered in our analysis.

This example demonstrates the thinking process involved in item model development. What we learn through the process of developing and analyzing item models can also be applied to discrete item writing. From an operational standpoint, the main goal of item modeling is to efficiently produce test items that are focused on assessing particular mathematical skills at specified levels of difficulty, but an additional benefit is that we learn more about which item features contribute to difficulty and discrimination.

Designing Item Models That Are Appropriate for the Purpose of the Assessment

From an assessment standpoint, an item model is most useful when it can generate instances that are informative with respect to the skill of interest. Item models may be designed for either high-stakes or low-stakes applications, but the criteria for successful design are somewhat different. For high-stakes applications like admissions testing programs, item models should be designed to generate instances that satisfy the psychometric criteria established by the testing program. For low-stakes applications like diagnostic assessments, item models should be designed to generate instances that capture common student misconceptions, so that they can be subsequently addressed through instruction.

In the previous section of this paper, we discussed how item models are conceptualized and developed. Once a model is complete, a sample of its instances is generated, administered, and evaluated with respect to the original goals for the model (in this case, to generate isomorphic instances).

In this section, we focus on the iterative aspect of item model development: After item models are developed, if they do not suit the purpose of the assessment for which they were designed, they are analyzed and revised. We describe the approach in the context of assessments with high-stakes outcomes, but it may be applied in contexts with low-stakes outcomes as well.

Designing item models that satisfy the psychometric criteria. For testing applications with high-stakes outcomes, an item model should generate instances that effectively discriminate

between low- and high-performances and are not easy to answer by guessing. Typically, in order to be included in a test, the discrimination parameter estimate (*a*) for an item must exceed a minimum acceptable value, and the "guessing" parameter estimate (*c*) must be lower than some maximum acceptable value. The item difficulty parameter estimate (*b*) may vary, but it should fall within a certain range (at least some examinees should be able to answer it, but not all examinees should be able to answer it). An item model that generates a large number of instances that meet these psychometric criteria therefore has a high *effective yield*, in the sense that it generates a large number of instances that may be incorporated into an assessment.

Incorporating Item Difficulty Predictions Into Item Model Development

In item model design, there are two alternative approaches that may be employed to systematically control difficulty. As mentioned earlier, Bejar (2002) makes a distinction between models that are designed to generate isomorphic instances versus models that are designed to generate variants. Isomorphs are instances that are equivalent in every way except with regard to their surface features. They are considered exchangeable; they share the same psychometric properties and problem structure, or schema. Variants within a model show systematic variation with regard to a particular characteristic and are generally not exchangeable. For example, a model may be designed to generate variants of different levels of difficulty (Bejar, 2002). The distinction between isomorphs and variants is the psychometric analog to Reed's conceptual distinction between isomorphic problems and similar problems, respectively (Reed, 1999).

A thorough specification of how item features influence difficulty is not easily obtained. In practice, it is challenging to design item models that generate either isomorphs or variants with predictable psychometric parameters. Experts' predictions about which factors contribute to item difficulty are not always accurate (e.g., Bejar, 1983; Camerer & Johnson, 1991; Nathan & Koedinger, 2000; Nathan, Koedinger, & Alibali, 2001; Nathan & Petrosino, 2003). In quantitative domains, alterations that seem insignificant to an expert can make instances easier or more difficult to the novice who may apply weaker, less general solution strategies. Weaker solution strategies can sometimes be less obvious to experts, a phenomenon Nathan et al. and Nathan and Petrosino referred to as the *expert blind spot*. Two items that appear equally difficult to the expert may appear quite different to the examinee. Many researchers in cognitive psychology and mathematics education have identified features that influence item difficulty in many quantitative areas. For example, the determinants of difficulty have been explored for

algebra problems (e.g., Embretson, 1995; Enright et al., 2002; Enright & Sheehan, 2002; Heffernan & Koedinger, 1998; Koedinger & MacLaren, 2002; Koedinger & Nathan, 2004; Mayer, Larkin, & Kadane, 1984; Sebrechts, Enright, Bennett, & Martin, 1996), proportional reasoning problems (e.g., Kaput & West, 1994 ; Karplus, Pulos, & Stage, 1983; Noelting, 1980; Vergnaud, 1980), arithmetic word problems (e.g., Enright et al., 2002; Kintsch & Greeno, 1985; Riley & Greeno, 1988), and quantitative reasoning items with more than one correct response (Bennett et al., 1999; Katz, Lipps, & Trafton, 2002).

In summary, item models are designed to generate instances that target the skills of interest. A model that generates isomorphic instances is particularly applicable to large-scale assessments for at least three reasons. First, the assessment designers want to ensure that a given skill of interest is being measured accurately and uniformly across instances. Bejar (2002) noted that the use of isomorphic instances ensures better test reliability and score precision. Second, the cognitive framework required to explain the psychometric variability for a broadly defined item model is necessarily complicated, and in an operational setting, it is not feasible to develop such a framework for each item model that must be developed. Finally, it may be possible to save on pretesting costs if instances generated from an item model inherit its statistical parameters.

Item Modeling as an Iterative Process

Research findings from mathematics education and cognitive psychology may inform how to design item models that generate instances with somewhat stable or at least predictable psychometric characteristics. In our experience, however, the factors that influence difficulty, discrimination, and guess rate can be particular to a specific situation, and can often only become evident in retrospect after the statistics from administered instances are reviewed. Bejar (1993), Bejar and Yocom (1991), and Embretson and Gorin (2001) emphasize that item models must be evaluated empirically, and since evaluation will often implicate revision, item models that generate instances appropriate for the assessment should be developed iteratively. We therefore recommend a cyclical approach (as illustrated in Figure 1) to item model development: Instances from a model are reviewed and analyzed, and the model is revised so that it generates instances that are more likely to meet the established validity and psychometric criteria.

Statistical and Cognitive Analysis Applied to Item Modeling: An Example

In this section, we describe how to analyze and revise an item model using the item model development loop. In this example, our goal was to design a model that would generate isomorphic instances, rather than variants. This section focuses on the conduct statistical analysis and conduct cognitive analysis steps in Figure 1 and corresponds to the model evaluation stage of the cognitive design system approach (Embretson, 1999; Embretson & Gorin, 2001).

Figure 7 shows a discrete instance generated from the linear-inequality, most-difficult item model (one of 16 models developed in this effort); the key is indicated.

The statement "$t - 3 \leq -1$ or $3 - t \geq 13$" is equivalent to which of the following?

A. $t \leq 2$ Key
B. $t \leq -10$
C. $-2 \leq t \leq 13$
D. $-9 \leq t \leq 3$
E. $-10 \leq t \leq 2$

Figure 7. **Example of linear inequality item model instance.**

Ten instances were generated from the linear inequality item model; each instance was presented as the sixth item on a 28-item unscored section. Each examinee saw only one instance from the model. In the data analysis sample, the average number of examinees per instance was 793. A 3PL IRT model (Birnbaum, 1968) with a scaling factor of 1.7 was fitted to the response scores. Table 2 shows the proportion correct and IRT parameter estimates for each of the model instances. The proportion correct across instances was highly variable (proportion correct ranges from .14 to .32). The IRT parameter estimates were also highly variable (*a*-parameter estimates range from .20 to 1.37; *b*-parameter estimates range from 2.52 to 4.10; and *c*-parameter estimates range from .05 to .26).

Table 2

Proportion Correct and Item Parameter Estimates for 10 Instances From a Linear Inequality Item Model

Instance	Performance class	Proportion correct	*Item parameter estimates*		
			a	*b*	*c*
7	Type 1	.14	0.40	3.46	0.05
8	Type 1	.14	0.40	3.47	0.06
6	Type 1	.15	0.89	3.31	0.14
1	Type 1	.16	1.37	2.52	0.14
2	Type 1	.16	1.16	2.55	0.13
4	Type 1	.16	0.99	2.80	0.14
9	Type 2	.26	0.20	3.89	0.05
5	Type 2	.30	0.20	3.12	0.06
10	Type 2	.30	0.49	4.10	0.26
3	Type 2	.32	0.59	2.74	0.25

The proportion correct data in Table 2 suggest that instances from the linear inequality model fall into two distinct performance classes. Instances with a lower proportion correct are labeled Type 1 and instances with a higher proportion correct are labeled Type 2.

Figure 8 has two panels: Each panel shows the item characteristic curves (ICCs) for 10 instances generated from an item model. The left panel shows the ICCs for an ideal item model; the right panel shows the ICCs for instances generated from the linear inequality model. The curves in the left panel almost overlap; there is very little area spanned by the curves. The instances generated from the ideal model are essentially isomorphic, because they share both a common problem structure (as defined by the item model) and similar psychometric properties. By contrast, as can be seen in the right panel of Figure 8, the ICCs for the linear inequality model do not appear to be isomorphic. Although the model was designed to generate similar instances, there appear to be at least two distinct performance classes.

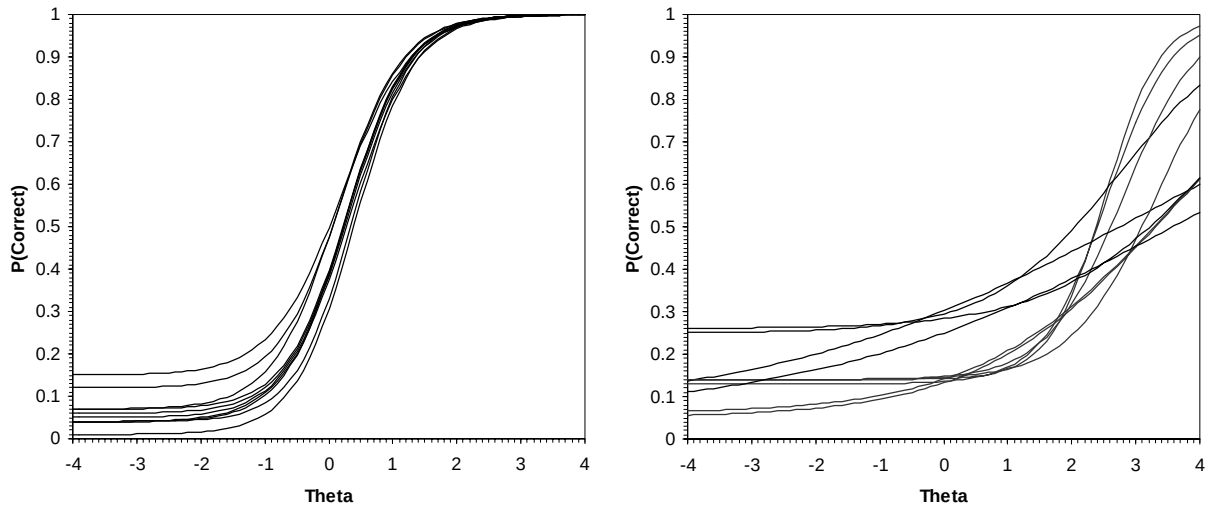

Figure 8. **ICC curves for models that generated instances with similar and variable psychometric properties.**

The two classes correspond to the performance types distinguished in Table 2 and are differentiated in Figure 9.

The data in Table 2 and the curves in Figure 9 both suggest that the linear inequality model may have generated at least two classes of instances. These are *emergent classes* in the sense that they are not distinct by design—our intention was to design a model that would generate isomorphic instances. This kind of result has been observed before. Meisner et al. (1993, p.11) noted "…apparent clusters of p-values (mean item scores) for some of the individual algorithms." The challenge at this point was to identify features of the instances that might account for the performance difference between the two types. As Meisner et al. did for algorithms with clustered p-values, we reviewed the linear inequality model for systematic variations in item structure and considered how the model might be revised. We will refer to instances that are associated with one of the two types as Type 1 and Type 2 instances, respectively. There are a number of variations that did not appear to be associated with performance, but Type 1 and Type 2 instances vary in one important way: They have slightly different sets of distractors.

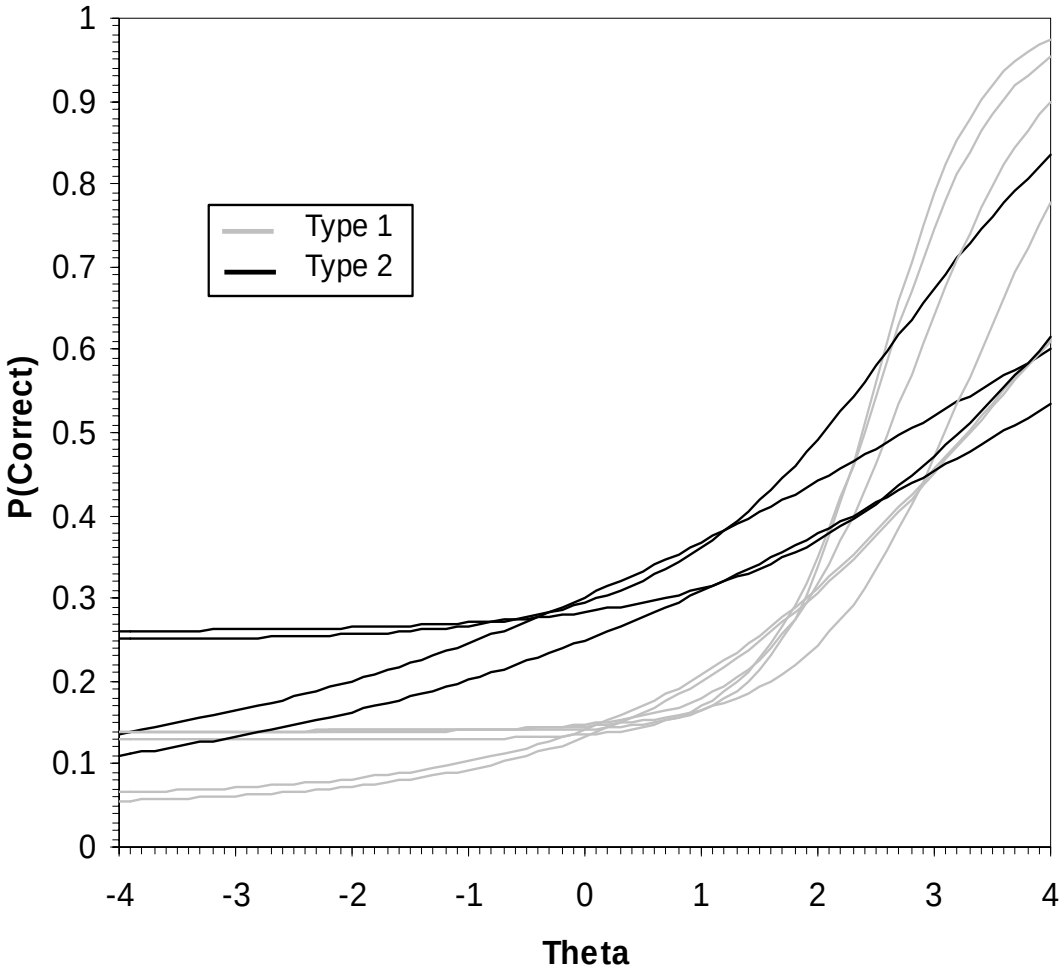

Figure 9. **Two types of ICCs for instances in the linear inequality model.**

As mentioned previously, in addition to modeling variables in the stem, a test developer also models the key and the distractors. We will refer to a variable or a combination of variables that represents a key or a distractor as a *key model* or a *distractor model*, respectively. For a particular item model, the collection of distractor models is called the *distractor model set*, and the collection of distractor models and the key model is the *option model set*. The number of distractor models in the set may exceed the number of distractors that appear in a given instance generated from the item model. When this occurs, only a subset of the distractor models is represented in any discrete instance. The linear inequality model included five different distractor models, but only four were represented in each instance.

Examples of Type 1 and Type 2 instances are shown in Figures 10 and 11. Each distractor has been labeled with the name of the distractor model that generated it, and equivalent

distractor models are designated with the same number. For example, Distractor Model 3 generated option C in the Type 1 instance (Figure 10) and option B in the Type 2 instance (Figure 11). Two of the distractor models (1 and 2) correspond to suspected misconceptions; the nature of the misconception is summarized in parentheses. The two types do not have equivalent distractor model sets. In particular, all of the Type 1 instances were generated using Distractor Models 1, 2, 3, and 4. All of the Type 2 instances were generated using Distractor Models 2, 3, 4, and 5.

Lower performance, mean proportion correct: .15

The statement "$t - 3 \leq -1$ or $3 - t \geq 13$" is equivalent to which of the following?

A. $t \leq 2$	Key model ($t \leq 2$ or $t \leq -10$)
B. $t \leq -10$	Distractor Model 2 (examinee interprets *or* as *and*)
C. $-2 \leq t \leq 13$	Distractor Model 3
D. $-9 \leq t \leq 3$	Distractor Model 4
E. $-10 \leq t \leq 2$	Distractor Model 1 (reversed inequality)

***Figure 10*. Example of Type 1 instance.**

Higher performance, mean proportion correct: .30

$x - 9 < -2$ or $4 - x > 19$

The statement above is equivalent to which of the following?

A. $-14 < x < 8$	Distractor Model 4
B. $-7 < x < 19$	Distractor Model 3
C. $x < -15$	Distractor Model 2 (examinee interprets *or* as *and*)
D. $x < 7$	Key model ($x < 7$ or $x < -15$)
E. $x < 19$	Distractor Model 5

***Figure 11*. Example of Type 2 instance.**

There are other differences between the two examples shown, including the stem format, the numbers used, and the order in which the distractor models appeared as distractors. Such differences were present within each type, however, and do not account for the performance difference between types.

The main difference between the two performance classes is that the Type 1 instances include Distractor Model 1, and the Type 2 instances replace Distractor Model 1 with Distractor Model 5. We suspected that Distractor Model 1 (which appears in the Type 1 instances only) corresponds to a popular misconception, and therefore generates attractive distractors. Distractor Model 5, however, which replaces Distractor Model 1 in the Type 2 instances, may not be as appealing—upon consideration, it does not seem as likely that an examinee will select an option corresponding to Distractor Model 5. It should be noted that this analysis is retrospective—without the benefit of item performance data, the instances from this model appear quite similar. There may be no strong a priori reason to believe that one distractor model is more compelling than another.

If our interpretation concerning distractor models is correct, then the substitution of Distractor Model 5 for Distractor Model 1 might account for the variability among the instances. In order to further explore this idea, we looked at the percentage of examinees that endorsed each option generated by a particular distractor model. If Distractor Model 1 generates attractive distractors, then distractors that correspond to it should claim a large percentage of the responses. The percentage of examinees that selected options corresponding to each distractor model is shown in Figure 12; lines with square markers represent the Type 1 instances and lines with circle markers represent the Type 2 instances. The figure suggests that Distractor Model 1 generated very attractive distractors; between 40–50% of examinees selected this option when it appeared, and it was much more popular than the key. Distractor Model 5, however, did not generate attractive distractors, perhaps because the model did not correspond to a popular misconception. Figure 12 also shows that Distractor Model 2, which corresponds to the *and* instead of the *or* interpretation of the stem, generates attractive distractors. A Distractor Model 2 option is not as popular as a Distractor Model 1 option, but it is still more popular than the key.

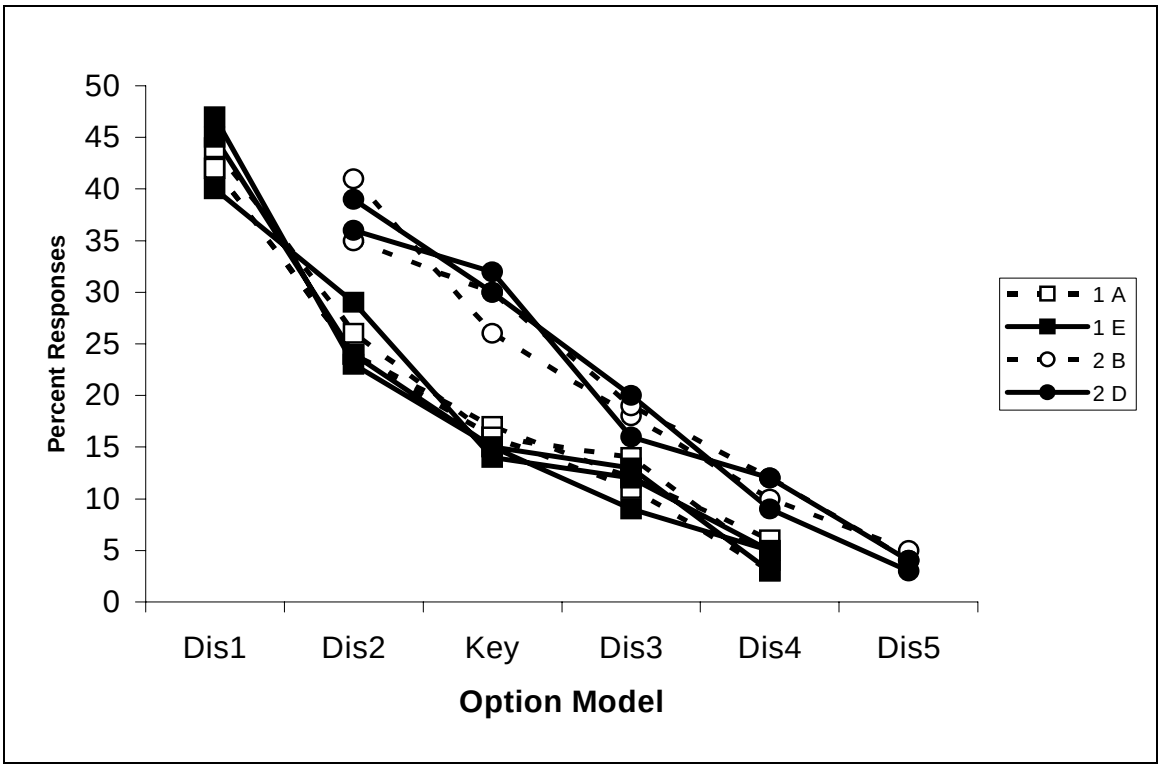

Figure 12. **Percentage of examinees that selected options generated by different option models on instances generated by the linear inequality model.**

For each of the two types, the options corresponding to the distractor models were presented in one of two sequences—half of the instances presented the options in one order, and the other half presented the options in the reverse order. This distinction is made on the graph in Figure 12; the dotted lines represent those instances for which the option models appeared in one order (A keys and B keys, for Type 1 and Type 2 instances, respectively), and the solid lines represent those instances for which the option models appeared in the reverse order (E keys and D keys, for Type 1 and Type 2 instances, respectively). None of the Type 1 curves overlap any of the Type 2 curves, but within each of the two types, curves corresponding to instances with different option orders are virtually indistinguishable. This suggests that it is the presence of a Distractor Model 1 option, and not its position, that affects performance. For this model, an option that represents Distractor Model 1 is likely to be endorsed by an examinee, regardless of where it appears among the options.

It is important to note that these findings are limited to instances from this particular model. In fact, we have found that for some models, performance on the instances is highly

associated with the relative positions of the key model and the distractor models. For example, some items do not have unique solutions, and the answer is found by identifying the option that satisfies the constraints specified in the stem. Such items typically require that examinees test each option, and it is likely that the position of the key will affect performance on instances generated from models that have this problem structure.

More generally, in different item models, different components may influence the variability of the instances. In the linear inequality example, even a slight difference in the composition of the distractor model set appears to play a very important role in determining the behavior of the instances. The potentially important role of distractor models for item generation has also been noted in other contexts. For example, Embretson and Gorin (2001) noted the important role of distractors as determinants of item difficulty for generating *object assembly* items. The stem of an object assembly item consists of scattered pieces. The task is to select the option that represents a whole object that is possible to assemble from the pieces shown in the stem. A similar observation has been made in the context of generating analytical reasoning items (Newstead, Bradon, Handley, Evans, & Dennis, 2002). Depending on the model, however, either features of the stem or features of the options may play the primary role.

Completing the Cycle

In addition to providing possible explanations for the variability among the linear inequality model instances, this analysis suggests a means for revising the model so that the instances are both isomorphic and more discriminating (i.e., have higher *a*-parameter estimates). This section focuses on revising the model and corresponds to the revise-item-model step from the item model development loop in Figure 1.

To generate instances with more comparable statistics, a good first step is to constrain the model to generate only Type 1 or only Type 2 instances. The Type 1 instances are preferable because, in general, the discrimination value (IRT parameter estimate *a*) is higher and the guessing value (IRT parameter estimate *c*) is lower. An item model that generates only Type 1 instances is likely to have a higher effective yield than an item model that generates only Type 2 instances, because the instances are more likely to satisfy testing programs' cutoff criteria for *a* and *c* values. To complete the item model development cycle, the linear inequality item model was constrained so that it would only generate Type 1 instances. This represents an attempt to

create an item model that generates instances with high discrimination values and low guessing values that are also similar in difficulty.

We administered 32 instances from the revised linear inequality model. As before, the options appear in one of two orders (16 instances were generated for each order). A close inspection of Figure 9 shows that even within the Type 1 instances, there are two instances that have lower discrimination values (the two flatter curves in Figure 9). Because there are only six Type 1 instances, and only two of them have lower discrimination values, it is difficult to ascertain which item model variables may be responsible for the difference. Once we have data from the revised item models, we may be able to identify which item model variables affect the discrimination rates of Type 1 instances.

For the linear inequality model example, we completed the cycle by revising the model in such a way that it should generate more closely isomorphic instances. This seemed straightforward to do in this case, because the item model variables contributing to performance are readily apparent. It should be mentioned that this is something of a singular case, however. Often, there are several item model variables that may affect how the instances behave, and often these variables are confounded. When this situation occurs, we take a different approach: We revise the model and systematically generate instances to isolate the potential effects of the confounded item model variables. When potential confounding is a concern, we attempt to generate variants at different levels of difficulty to better understand the influence of different item model variables. It should be noted that for a small number of models, the results may be so difficult to interpret that it is not possible to systematically generate instances even to test candidate hypotheses. In this event, we recommend redesigning the model from scratch. This involves reexamining the source item (and perhaps several others items like it) to determine what skills the item(s) were originally designed measure. It may be that the model instances vary in their parameters because they actually test slightly different skills. Once the underlying skills have been reevaluated, a new item model may be designed that is intended to preserve the rationale behind the development of the original source item(s).

Caveats and Recommendations

There are limitations to the approach we describe here. One obvious limitation is that it is time-consuming. Another limitation is that this approach is a model-fitting exercise; we are trying to make a generalization about a large set of instances (all the instances in a model) based

on a relatively small set of instances. It may be possible to develop an approach for sampling instances, however, since the instances vary in a systematic fashion. Another potential risk with this approach is that in the process of revising models, it is possible to introduce additional variables that may have unforeseen effects on performance. When this occurs, we try to compartmentalize models, so that only one new variable is introduced in each new model.

In our discussion of the linear inequality model, it probably became clear that apparently small differences in the composition of the instances can have very large effects on performance, and we have observed that this is often the case with other item models as well. In order to develop item models that generate isomorphic instances, it may be necessary to constrain the models to a high degree. While this approach may help us to control the psychometric parameters, the downside is that it may significantly limit the variability of the instances generated by the model. As noted by Messick (1994), two major threats to construct validity include construct-irrelevant variance and construct underrepresentation. In the context of item model development, this presents a challenge: How do we control difficulty and discrimination and sufficiently represent the construct at the same time? If a set of item models is going to be practically useful, it needs to span a reasonable amount of content. Further, item models that generate instances that are too similar may be coachable.

Fortunately, we believe there is a practical approach to item model development that avoids both threats to construct validity. Item models can be designed in a modular fashion, as parts of larger model families. It is not much more difficult to create a family of narrowly defined models than it is to create a single, broadly defined model, and defining families of isomorphic models gives us the capacity to control the psychometric parameters of the instances. Item models within a family that perform similarly may always be consolidated later on. It may be more appropriate to think of the item model family, rather than the item model, as the essential unit of development. Across families, it is possible to span many content areas. We therefore recommend that item models be developed as parts of families when possible.

Conclusions and Questions for Further Research

Questions for further research target improving the iterative approach to item model development, so that it is both more effective and more efficient. There are opportunities for improvement at almost every stage of the process. First, what strategies might be used at the outset, so that new item models are more likely to generate instances that suit the purpose of the

assessment? Although the results we have found so far are quite specific to the content from a particular model, some results may be generalizable. For example, as a general rule it may be extremely important to attend to distractor sets and how comparable they are across instances. Eventually, we may be able to compile a knowledge base consisting of both general and content-specific information that test developers may consult prior to designing a new item model. At the very least, we should be able to define a set of best-practice guidelines for item model design, so that the effective yield is as high as possible.

We may be able to make improvements to the sample-instances stage. The iterative approach is based on the assumption that it is possible to make generalizations about a model based on a small sample of instances generated by the model. It is not clear how many instances should be sampled, or how to select a representative sample. It also is not clear that the sample should be the same size for every model. In addition, samples may be generated randomly or systematically. Typically, when we first test a model, the sample consists of instances that have been generated randomly, with constraints to ensure that instances are not repeated. Once the model is revised, the sample usually consists of instances that have been generated systematically, in order to test the hypotheses of interest. Both methods for sampling instances are important—the former is important in order to identify unanticipated effects of item model variables; the latter is important to test candidate hypotheses. We should explore at what point in the process random versus systematic generation is preferable, or if both are included, how to define an optimal set of instances for testing. Finally, it may be possible to develop metrics that indicate when a model generates instances that meet the criteria established at the outset. These criteria will vary, depending on the purpose of the assessment for which a model is designed. But there should be objective measures of whether a model is sufficiently improved or whether it is likely to benefit from another iteration of the development cycle.

References

Bejar, I. I. (1983). Subject matter experts' assessment of item statistics. *Applied Psychological Measurement, 7*(3), 303–310.

Bejar, I. I. (1993). A generative approach to psychological and educational measurement. In N. Frederiksen (Ed.), *Test theory for a new generation of tests* (pp. 323–357). Hillsdale, NJ: Lawrence Erlbaum Associates.

Bejar, I. I. (1996). *Generative response modeling: Leveraging the computer as a test delivery medium* (ETS RR-96–13). Princeton, NJ: ETS.

Bejar, I. I. (2002). Generative testing: From conception to implementation. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 199–218). Mahwah, NJ: Lawrence Erlbaum Associates.

Bejar, I. I., Lawless, R. R., Morley, M. E., Wagner, M. E., Bennett, R. E., & Revuelta, J. (2002). *A feasibility study of on-the-fly item generation in adaptive testing* (GRE Board Professional Rep. No. 98–12P). Princeton, NJ: ETS.

Bejar, I. I., & Yocom, P. (1991). A generative approach to the modeling of isomorphic hidden-figure items. *Applied Psychological Measurement, 15*(2), 129–137.

Bennett, R. E., Morley, M., Quardt, D., Rock, D. A., Singley, M. K., Katz, I. R., & Nhouyvanisvong, A. (1999). Psychometric and cognitive functioning of an under-determined computer-based response type for quantitative reasoning. *Journal of Educational Measurement, 36,* 233–252.

Birnbaum, A. (1968). Some latent trait models. In F.M. Lord & M.R. Novick (Eds.), *Statistical theories of mental test scores* (pp. 397–424). Reading, MA: Addison-Wesley.

Camerer, C. F., & Johnson, E. J. (1991). The process-performance paradox in expert judgment: How can experts know so much and predict so badly? In J. Smith & K. A. Ericsson (Eds.), *Toward a general theory of expertise: Prospects and limits* (pp. 195–217). New York: Cambridge University Press.

Embretson, S. E. (1995). A measurement model for linking individual learning to processes and knowledge: Application to mathematical reasoning. *Journal of Educational Measurement, 32*(3), 277–294.

Embretson, S. E. (1999). Generating items during testing: Psychometric issues and models. *Psychometrika, 64*(4), 407–433.

Embretson, S., & Gorin, J. (2001). Improving construct validity with cognitive psychology principles. *Journal of Educational Measurement, 38*(4), 343–368.

Enright, M. K., Morley, M., & Sheehan, K. M. (2002). Items by design: The impact of systematic feature variation of item statistical characteristics. *Applied Measurement in Education, 15*(1), 49–74.

Enright, M. K., & Sheehan, K. M. (2002). Modeling the difficulty of quantitative reasoning items: Implications for item generation. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 129–157). Mahwah, NJ: Lawrence Erlbaum Associates.

Heffernan, N. T., & Koedinger, K. (1998). A developmental model for algebra symbolization: The results of a difficulty factors assessment. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society,* (pp. 484–489). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hively, W., Patterson, H. L., & Page, S. H. (1968). A "universe-defined" system of arithmetic achievement tests. *Journal of Educational Measurement, 5*(4), 275–290.

Irvine S. H. (2002). The foundations of item generation for mass testing. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 3–34). Mahwah, NJ: Lawrence Erlbaum Associates.

Kaput, J. J., & West, M. M. (1994). Missing-value proportional reasoning problems: Factors affecting informal reasoning patterns. In G. Harel & J. Confrey (Eds.), *The development of multiplicative reasoning in the learning of mathematics* (pp. 235–287). Albany, NY: State University of New York Press.

Karplus, R., Pulos, S., & Stage, E. K. (1983). Proportional reasoning of early adolescents. In R. Lash & M. Landau (Eds.), *Acquisition of mathematics concepts and processing* (pp. 45–86). New York: Academic Press.

Katz, I., Lipps, A., & Trafton, J. (2002). *Factors affecting difficulty in the generating examples item type* (GRE Board Professional Rep. No. 97–18P). Princeton, NJ: ETS.

Kintsch, W., & Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological Review, 92*(1), 109–129.

Koedinger, K. R., & Nathan, M. J. (2004). The real story behind story problems: Effects of representations on quantitative reasoning. *The Journal of the Learning Sciences, 13*(2), 129–164.

Koedinger, K. R., & MacLaren, B. A. (2002). *Developing a pedagogical domain theory of early algebra problem solving* (CMU-HCII Tech. Rep. No. 02–100). Pittsburgh: PA: Carnegie Mellon University.

LaDuca, A., Staples, W. I., Templeton, B., & Holzman, G. B. (1986). Item modeling procedure for constructing content-equivalent multiple-choice questions. *Medical Education, 20*(1), 53–56.

Mayer, R. E., Larkin, J. H., & Kadane, J. B. (1984). A cognitive analysis of mathematical problem-solving ability. In R. J. Sternberg (Ed.), *Advances in the psychology of human intelligence* (pp. 231–273). Hillsdale, NJ: Lawrence Erlbaum Associates.

Meisner, R. M. Luecht, R., & Reckase, M. D. (1993). *The comparability of the statistical characteristics of test items generated by computer algorithms* (ACT Research Rep. Series No 93-9). Iowa City, IA: The American College Testing Program.

Messick, S. (1994). *Standards-based score interpretation: Establishing valid grounds for valid inferences* (ETS RR-94–57). Princeton, NJ: ETS.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2002). On the roles of task model variables in assessment design. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 97–128): Mahwah, NJ: Lawrence Erlbaum Associates.

Morley, M. E., Bridgeman, B., & Lawless, R. R. (2004). *Transfer between variants of quantitative items* (GRE Board Rep. No. 00–06R). Princeton, NJ: ETS.

Nathan, M. J., & Koedinger, K., R. (2000). Teachers' and researchers' beliefs about the development of algebraic reasoning. *Journal for Research in Mathematics Education, 31*(2), 168–190.

Nathan, M. J., Koedinger, K. R., & Alibali, M. W. (2001). Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In L. Chen et al. (Eds.), *Proceeding of the Third International Conference on Cognitive Science* (pp. 644–648). Beijing, China: USTC Press.

Nathan, M. J., & Petrosino, A. (2003). Expert blind spot among preservice teachers. *American Educational Research Journal, 40*(4), 905–928.

Newstead, S., Bradon, P., Handley, S., Evans, J., & Dennis, I. (2002). Using the psychology of reasoning to predict the difficulty of analytical reasoning problems. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 35–51). Mahwah, NJ: Lawrence Erlbaum Associates.

Noelting, G. (1980). The development of proportional reasoning and the ratio concept: Part I - Differentiation of stages. *Educational Studies in Mathematics, 11*(2), 217–253.

Reed, S. K. (1999). *Word problems: Research and curriculum reform*. Mahwah, NJ: Lawrence Erlbaum Associates.

Riley, M. S., & Greeno, J. G. (1988). Developmental analysis of understanding language about quantities and of solving problems. *Cognition and Instruction, 5*(1), 49–101.

Sebrechts, M. M., Enright, M., Bennett, R. E., & Martin, K. (1996). Using algebra word problems to assess quantitative ability: Attributes, strategies, and errors. *Cognition and Instruction, 14*(3), 285–343.

Shute, V. J., Graf, E. A., & Hansen, E. (2005). Designing adaptive, diagnostic math assessments for individuals with and without disabilities. In L. PytlikZillig, R. Bruning, & M. Bodvarsson (Eds.), *Technology-based education: Bringing researchers and practitioners together* (pp. 169–202). Greenwich, CT: Information Age Publishing.

Singley, M. K., & Bennett, R. E. (2002). Item generation and beyond: Applications of schema theory to mathematics assessment. In S. H. Irvine (Ed.), *Item generation for test development* (pp. 361–384). Mahwah, NJ: Lawrence Erlbaum Associates.

Sinharay, S., & Johnson, M. (2005). *Analysis of data from an admissions test with item models* (ETS RR-05-06). Princeton NJ: ETS.

Steffen, M., Graf, E. A., Levin, J., & Robin, F. (2005). *An investigation of the psychometric equivalence of quantitative isomorphs: Phase 1*. Manuscript in preparation.

Vergnaud, G. (1980). Didactics and acquisition of “multiplicative structures” in secondary schools. In W. F. Archenhold, A. Orton, R. H. Driver, & C. Wood-Robinson (Eds.), *Cognitive development research in science and mathematics* (pp. 190–201). Leeds, England: University of Leeds.

Notes

[1] As an example, if the base item showed a figure of a triangle and asked for the perimeter, the appearance variant would show the same figure but would ask for the area.